Visual Basic : เทคนิคแดร็กแอนด์ดร็อบแบบง่าย ๆ

เทคนิคแดร็กแอนด์ดร็อบเป็นเทคนิคที่นิยมใช้ในแอพพลิเคชันบนวินโดวส์ เป็นการเขียนโปรแกรมแบบง่าย ๆ

กลับมาเจอกันอีกครั้งกับเทคนิคการเขียนโปรแกรมด้วยวิชวลเบสิก ซึ่งมีลูกเล่นอีกมากมายที่จะใช้ได้ ปัจจุบันแอพพลิเคชันบนวินโดวส์ส่วนใหญ่มักจะสนับสนุนการทำงานแบบแดร็กแอนด์ดร็อบ เทคนิคการเขียนโปรแกรมสนับสนุนแดร็กแอนด์ดร็อบไม่ใช่เรื่องยากเลย เพราะในวิชวลเบสิกมีโพรซีเดอร์ที่เตรียมมาให้แล้ว ในบทความตอนนี้จะแสดงวิธีการเขียนโดยจะนำเอาพื้นฐานเดิมที่กล่าวไปแล้ว คือ ไดรฟ์ลิสต์ ไดเรคทอรีลิสต์ และไฟล์ลิสต์กลับมาใช้อีกครั้งหนึ่ง แต่จะแก้ไขโพรซีเดอร์บางอัน นอกจากนี้ก็จะแสดงเทคนิคการเขียนโปรแกรมบางอันที่ผู้อ่านจะสามารถนำไปใช้ได้ด้วย

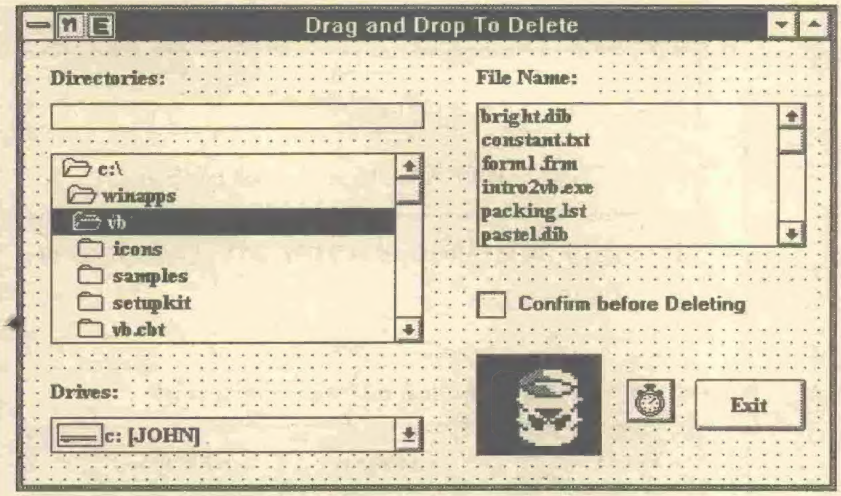
ลากไฟล์ไปเผา

สำหรับแอพพลิเคชันที่จะใช้แสดงวิธีการเขียนโปรแกรมให้สนับสนุนแดร็กแอนด์ดร็อบในบทความนี้คือ แอพพลิเคชันจะประกอบด้วยคอนโทรลไดรฟ์ ไดเรคทอรี ไฟล์ลิสต์บ็อกซ์ ผู้ใช้จะสามารถแดร็กไฟล์จากไฟล์ลิสต์บ็อกซ์เพื่อนำไปดรอบบนไอคอนถังขยะเพื่อจะลบไฟล์นั้นได้

ก่อนอื่นก็คงต้องเตรียมสร้างฟอร์มที่จะใช้เป็นหน้าต่างหลักของส่วนประกอบของฟอร์มในแอพพลิเคชันนี้ค่อนข้างมากสักหน่อย จากรูปที่ 1 จะแสดงคอนโทรลที่เป็นส่วนประกอบทั้งหมด ได้แก่ ไดรฟ์ ไดเรคทอรี ไฟล์ลิสต์บ็อกซ์ เลเบลต่าง ๆ อิมเมจทั้งหมด 5 รูป, ไทม์เมอร์ และคอมมานด์บัททอน

ค่าเริ่มต้นของฟอร์มจะคล้ายกับแอพพลิเคชันก่อน ๆ สำหรับ Property ที่ควรกำหนดก็เป็นชื่อฟอร์ม Caption ชนิดขอบ ตำแหน่งและขนาดหน้าต่าง ไม่ให้มีบัททอนใช้ขยายหน้าต่าง ในฟอร์ม DragDel นี้จะมีคอนโทรลแยกออกเป็น 5 กลุ่มได้แก่

- กลุ่มของลิสต์บ็อกซ์ ที่ใช้จัดการเกี่ยวกับการหาไฟล์ ค่าเริ่มต้นของคอนโทรลแต่ละอันได้แสดงอยู่ในตารางที่ 2 แล้ว สำหรับ Property ที่ต้องกำหนดค่าก็เช่นฟอนต์ที่ใช้จะเลือกเป็นฟอนต์ทรูไทพ์ Times New Roman ขนาด 9.75 นอกจากนี้ผู้เขียนยังฝากไอคอนที่จะใช้เวลาแดร็กแอนด์ดร็อบไว้กับ Property DragIcon ของ drvList และ



รูปที่ 1 แสดงคอนโทรลที่เป็นส่วนประกอบของฟอร์ม DragDel

ตารางที่ 1 แสดงค่าเริ่มต้นของฟอร์ม DragDel

Property	ค่าเริ่มต้น
Name	DragDel
BorderStyle	1-Fixed Single
Caption	Drag and Drop To Delete
Height	4665
Left	2190
MaxButton	False
ScaleHeight	4260
ScaleWidth	7065
Top	2040
Width	7185

รูปที่ 2 แสดงภาพไอคอนในไฟล์ DRAG1PG.ICO, DROP1PG.ICO

ตารางที่ 2 แสดงค่าเริ่มต้นของคอนโทรลกลุ่มลิสต์บ็อกซ์ต่าง ๆ

Object	Property	ค่าเริ่มต้น
DriveListBox	Name	drvList
	DragIcon	DRAG1PG.ICO
	FontName	Times New Roman
	FontSize	9.75
DirListBox	Name	dirList
	DragIcon	DROP1PG.ICO
	FontName	Times New Roman
	FontSize	9.75
FileListBox	Name	fileList
	FontName	Times New Roman
	FontSize	9.75

dirList โดยใช้ภาพจากไฟล์ DRAG-1PG.ICO เป็นไอคอนขณะแดร็กและ DROP1PG.ICO แทนขณะดร็อบ (ดูรูปที่ 2)

- กลุ่มของเลเบลต่าง ๆ ซึ่งใช้อธิบายความหมายของคอนโทรล โดยจะต้องกำหนดค่าของ Caption ใหม่ (ดูตารางที่ 3 และรูปที่ 1) สำหรับฟอนต์ที่เลือกใช้ก็เป็น Times New Roman ขนาด 9.75 ทั้งหมด

- กลุ่มของอิมเมจทั้ง 2 รูป ได้แก่ คอนโทรล trash และ fire ซึ่งเป็นภาพที่ใช้เป็นบริเวณสำหรับดร็อบไฟล์ โดยคอนโทรล trash จะเป็นภาพถังขยะ ใช้ภาพจากไฟล์ TRASH5.ICO และคอนโทรล fire เป็นภาพถังขยะแบบเดียวกันแต่มีเปลวไฟจากไฟล์ TRASH-6.ICO (ดูรูปที่ 3) คอนโทรลทั้งสองจะต้องวางซ้อนในตำแหน่งเดียวกันให้

Name	ค่าเริ่มต้น
lblDrives	Drives:
lblCurDir	" "
lblDirectories	Directories:
lblFilename	File Name:
confirm	Confirm before deleting

ตารางที่ 3 แสดงค่าเริ่มต้นของ Property Caption ทั้งหมด

รูปที่ 3 แสดงภาพคอนโทรล trash และ fire

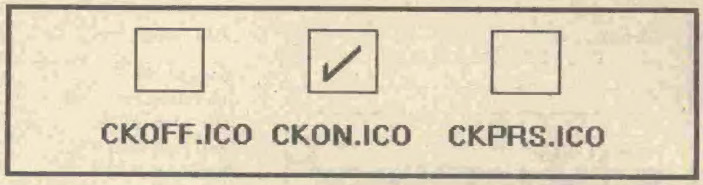
ตารางที่ 4 แสดงค่าเริ่มต้นของคอนโทรล Image สองอันที่ซ้อนกัน

Name	Property	ค่าเริ่มต้น
trash	Height	975
	Left	3960
	Picture	TRASH5.ICO
	Strech	True
	Top	3000
	Width	1095
fire	Height	975
	Left	3960
	Picture	TRASH6.ICO
	Strech	True
	Top	3000
	Width	1095

คอนโทรลที่	Property	ค่าเริ่มต้น
1	Height	255
	Index	0
	Left	3960
	Picture	CKOFF.ICO
	Strech	True
	Top	2400
	Visible	False
	Width	255
2	Index	1
	Picture	CKON.ICO
	Strech	True
	Visible	True
3	Index	2
	Picture	CKPRS.ICO
	Strech	True
	Visible	False

ทับกันสนิท ดังนั้นจึงต้องกำหนดให้ค่า Height, Left, Top, Width ของทั้งสองภาพเท่ากัน สาเหตุที่ผู้เขียนกำหนดให้ภาพทั้งสองวางซ้อนกันก็เนื่องจากผู้เขียนจะทำภาพของถังขยะให้มีเปลวไฟลุกเป็นระยะ ๆ แบบแอนนิเมชัน นอกจากนี้ก็ต้องกำหนดให้ Property Strech มีค่า True ด้วยเพื่อให้ภาพยืดขนาดออกเต็มในบริเวณที่เรากำหนดไว้

- กลุ่มอิมเมจอีก 3 รูปที่ซ้อนกัน ซึ่งเราใช้สร้างเป็น Custom Check Box (ดูรูปที่ 4) คือเช็คบ็อกซ์ที่สร้างขึ้นนี้



รูปที่ 4 แสดงภาพไอคอนในไฟล์ CKOFF.ICO, CKON.ICO และ CKPRS.ICO

ตารางที่ 6 แสดงค่าเริ่มต้นของคอนโทรลที่เหลือในฟอร์ม

Object	Property	ค่าเริ่มต้น
Timer	Interval	1000
CommandButton	Name	cmdExit
	Caption	Exit
	FontName	Times New Roman
	FontSize	9.75

ตารางที่ 5 แสดงค่าเริ่มต้นของคอนโทรลอะเรย์ Image ชื่อ checked 3 รูป

จะใช้เครื่องหมายถูกแทน 'X' เพื่อเช็คค่า คอนโทรลอะเรย์ที่สร้างนี้เป็นชนิด Image ชื่อ checked อิมเมจทั้งสามจะต้องมีขนาดเท่ากันและซ้อนกันพอดี ดังนั้นค่า Height, Left, Top, Width ต้องกำหนดให้เท่ากันหมด อิมเมจทั้งสามรูปจะนำมาจากไฟล์ CKOFF.ICO, CKON.ICO, CKPRS.ICO ตามลำดับ โดยกำหนดให้ Property Visible ของ checked(1) เป็น True ส่วนของ checked(0), checked(2) เป็น False

- คอนโทรล Timer และ CommandButton โดยเราจะใช้ Timer เพื่อควบคุมการสร้างภาพแอนนิเมชันของถังขยะให้มีเปลวไฟ โดยกำหนดค่า Interval ให้ทำงานทุก ๆ 1 วินาที ส่วนบัททอน cmdExit จะกำหนดค่า Caption เป็น Exit และเลือกใช้เช่นเดียวกับคอนโทรลอื่น ๆ

เริ่มเขียนโพรซีเดอร์ง่าย ๆ กันก่อน

ก่อนที่จะเริ่มเขียนโพรซีเดอร์สำหรับเหตุการณ์ต่าง ๆ ก็จะต้องประกาศตัวแปรแบบโกลบอลที่จะใช้ในโปรแกรมก่อน โดยจะเขียนในส่วน general ดังนี้

```
Option Explicit
Dim myPath, myFile As String
Dim ConfirmFlag As Integer
Const MB_ICONEXCLAMATION = 48
```

ตัวแปร myFile ชนิดสตริงใช้เก็บชื่อไฟล์ที่ผู้ใช้เลือกและต้องการจะลบ ส่วนตัวแปร ConfirmFlag ชนิด Integer ใช้ตรวจสอบว่าจะมีไดอะล็อก-บ็อกซ์เพื่อยืนยันก่อนจะลบไฟล์ที่เลือกแล้วก่อนหรือไม่

จากนั้นเราจะเริ่มเขียนโพรซีเดอร์สำหรับเหตุการณ์ต่าง ๆ ของไดรฟ์ลิสต์-บ็อกซ์และไดเรคทอรีลิสต์บ็อกซ์ก่อน เริ่มจาก drvList_Change( ) ซึ่งรับผิดชอบเมื่อผู้ใช้เลือกช่องดิสก์ไดรฟ์ใหม่ โดยเราต้องเขียนโปรแกรมคอยรับผิดชอบความผิดพลาดกรณีที่ไม่ได้ปิดฝาดิสก์หรือไม่ได้ใส่ดิสเกตต์ด้วย จากนั้นก็เป็น dirList_Change( ) ที่นำค่า Path ของ dirList มากำหนดให้ fileList และแก้ไขค่า Caption ของ lblCurDir ให้แสดงชื่อไดเรคทอรีปัจจุบันตามรูปที่ 5

```
Sub drvList_Change ()
Dim Msg As String
   On Error GoTo DriveError
   dirList.Path = drvList.Drive
   Exit Sub
DriveError:
   Msg = Error$(Err)
   MsgBox Msg, MB_ICONEXCLAMATION, "Drag and Drop"
   Exit Sub
End Sub

Sub dirList_Change ()
   fileList.Path = dirList.Path
   lblCurDir.Caption = dirList.Path
End Sub
```

รูปที่ 5

โพรซีเดอร์ต่อไปที่จะกล่าวถึงคือโพรซีเดอร์ที่รับผิดชอบต่อเหตุการณ์ Timer ซึ่งผู้เขียนตั้งใจที่จะใช้แสดงแอนนิเมชันของภาพถังขยะมีเปลวไฟ Timer นี้จะถูกเรียกมาทำงานทุก ๆ 1 วินาที ตามค่า Interval ที่ตั้งไว้ ผู้เขียนกำหนดให้ถังขยะมีเปลวไฟติดทุก 5 วินาที โดยไฟจะลุกอยู่นาน 1 วินาที ดังนั้นจึงต้องใช้ตัวแปร Counter เป็นตัวนับเวลา เมื่อถึงเวลาที่ต้องการก็จะกำหนดให้มีการแสดงภาพถังขยะติดไฟ โดยกำหนดให้ค่า Property fire.Visible เป็น True ส่วน trash.Visible ที่เป็นภาพถังไม่มีไฟจะกำหนดให้มีค่า False และเมื่อเวลาผ่านไป 1 วินาทีจึงค่อยเซตค่ากลับดังนี้

```
Sub Timer1_Timer( )
Static Counter As Integer
      Counter = Counter + 1
      If Counter = 5 Then
            trash.Visible = False
            fire.Visible = True
      End If
      If Counter > 5 Then
            Counter = 0
            trash.Visible = True
            fire.Visible = False
      End If
End Sub
```

โพรซีเดอร์อื่น ๆ ที่จะกล่าวถึงตอนนี้ก็คือ Form_Load( ) ซึ่งจะใช้กำหนดค่าเริ่มต้นของตัวแปรต่าง ๆ เช่น ConfirmFlag มีค่าเป็น 1 หมายถึงเริ่มต้นผู้ใช้ต้องยืนยันก่อนลบไฟล์ เป็นต้น นอกจากนี้ก็จะเรียกซับรูทีน ExplodeForm ที่ทำหน้าที่แสดงเอฟเฟกท์ขณะที่กำลังเปิดหน้าต่างของโปรแกรมให้มีลักษณะระเบิดออกจากจุดกลางหน้าต่างแล้วขยายออกจนเท่ากับขนาดหน้าต่าง, cmdExit_Click( ) ใช้สำหรับจบการทำงานของโปรแกรม รายละเอียดของโพรซีเดอร์ทั้งสองมีดังนี้

```
Sub Form_Load( )
      ConfirmFlag = 1
      lblCurDir.Caption = dirList.Path
      fileList.Pattern = "*.*"
      ExplodeForm DragDel
End Sub
Sub cmdExit_Click( )
      End
End Sub
```

โพรซีเดอร์กลุ่มที่รับผิดชอบต่อการแดร็กแอนด์ดร็อบ

การเขียนโปรแกรมให้สนับสนุนแดร็กแอนด์ดร็อบนั้นจะต้องเกี่ยวข้องกับเหตุการณ์อยู่ 3 อัน คือ MouseDown, DragDrop และ DragOver สำหรับในแอพพลิเคชันนี้กำหนดให้ผู้ใช้สามารถแดร็กชื่อไฟล์จากคอนโทรล

```
Sub fileList_MouseDown (Button As Integer, Shift As Integer, <--
X As Single, Y As Single)
  fileList.DragIcon = dirList.DragIcon
  fileList.Drag
End Sub
```

รูปที่ 6

```
Sub trash_DragOver (Source As Control, X As Single, Y As     <--
Single, State As Integer)
  Select Case State
  Case 0
    fileList.DragIcon = drvList.DragIcon
  Case 1
    fileList.DragIcon = dirList.DragIcon
  End Select
End Sub
```

รูปที่ 7

ไฟล์ลิสต์บ็อกซ์ fileList เพื่อมาดร็อบบนบริเวณคอนโทรลอิมเมจ trash ซึ่งเหตุการณ์ทั้งหมดนี้จะเริ่มจากผู้ใช้ต้องคลิ๊กเม้าส์บนชื่อไฟล์ที่ต้องการในไฟล์ลิสต์บ็อกซ์ ดังนั้นโพรซีเดอร์เหตุการณ์แรกที่ผู้อ่านต้องเขียนก็คือ fileList__MouseDown( ) (ดังแสดงในรูปที่ 6)

จะเห็นว่าในโปรแกรมจะกำหนดให้ DragIcon ของ fileList มีค่าเป็นเท่ากับ DragIcon ที่ฝากไว้กับ dirList ซึ่งจะมีผลทำให้เคอร์เซอร์ของเม้าส์เปลี่ยนไปเป็นรูปตามไอคอน DRAG-1PG.ICO (ดูรูปที่ 2) หลังจากนั้นก็จะเรียก fileList.Drag ซึ่งเป็น Method ของคอนโทรล ซึ่งเป็นการบอกว่าจะเริ่มต้นใช้แดร็กแอนด์ดร็อบโอเปอเรชัน

ขณะที่กำลังทำแดร็กแอนด์ดร็อบโอเปอเรชัน เหตุการณ์ DragOver จะเกิดขึ้นตลอดเวลา ดังนั้นจึงสามารถใช้เหตุการณ์นี้คอยตรวจสอบการทำงานได้ เช่นในตัวอย่างนี้เราจะใช้ DragOver ตรวจว่าผู้ใช้ได้แดร็กคอนโทรลมาอยู่ภายในดร็อบเอเรียลแล้วหรือยัง โดยตรวจสอบจากตัวแปร State ซึ่งถ้ามีค่าเป็น 0 ก็แสดงว่าคอนโทรลถูกแดร็กมาอยู่ในดร็อบเอเรียล เราก็จะเปลี่ยนให้เม้าส์เคอร์เซอร์เปลี่ยนเป็นรูป DROP-1PG.ICO (ดูรูปที่ 2) ซึ่งเราฝากไว้กับ drvList.DragIcon แต่ถ้า State มีค่าเป็น 1 แสดงว่าคอนโทรลถูกแดร็กออกนอกดร็อบเอเรียล เราก็ต้องเปลี่ยนให้ DragIcon มีค่าเหมือนตอนเริ่มต้น (ดังแสดงในรูปที่ 7)

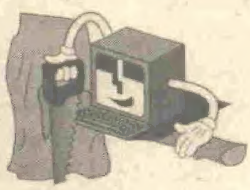
```
Sub trash_DragDrop (Source As Control, X As Single, Y As Single)
Dim Msg As String
Dim Response As Integer
Const ID_YES = 6
Const MB_YESNO = 4

  On Error GoTo DelError
  myPath = dirList.Path
  If Right(myPath, 1) <> "\" Then
    myPath = myPath + "\"
  End If
  myFile = myPath + fileList.FileName
  If ConfirmFlag = 1 Then
    Msg = "Do you want to delete the file " +          <--
UCase(myFile) + " ?"
    Response = MsgBox(Msg, MB_YESNO + MB_ICONEXCLAMATION, <--
"Drag and Drop")
    If Response = ID_YES Then
      Kill myFile
      fileList.Refresh
    End If
  Else
    Kill myFile
    fileList.Refresh
  End If
  Exit Sub

DelError:
  Msg = Error$(Err)
  MsgBox Msg, MB_ICONEXCLAMATION, "Drag and Drop"
  Exit Sub
End Sub
```

รูปที่ 8

หลังจากที่แดร็กแอนด์ดร็อบโอเปอเรชันสิ้นสุด คือเมื่อผู้ใช้ดร็อบคอนโทรลแล้วจะมีเหตุการณ์ DragDrop เกิดขึ้น ซึ่งเราจะใช้เหตุการณ์นี้เพื่อทำการลบไฟล์ที่แดร็กมา เริ่มจากเราจะต้องสร้างชื่อไฟล์ที่รวมพาธเพื่อจะนำไปใช้ลบไฟล์ โดยจะนำค่า dirList.Path มารวมกับ fileList.FileName จากนั้นก็ตรวจว่าจะต้องยืนยันการลบไฟล์ก่อนหรือไม่ โดยตรวจค่า ConfirmFlag ถ้าต้องยืนยันก็จะแสดงเมสเสจบ็อกซ์ให้ผู้ใช้เลือก หลังจากที่ลบไฟล์ด้วย Kill แล้ว เราจะต้องปรับให้ไฟล์ลิสต์บ็อกซ์แสดงค่าใหม่โดยเรียก Refresh กรณีที่ไม่สามารถลบไฟล์ได้ การทำงานจะข้ามไปที่ DelError ทันที (ดูรูปที่ 8)

โพรซีเดอร์สำหรับ Custom Checked Box

เช็คบ็อกซ์ของวิชวลเบสิกนั้นจะใช้เครื่องหมาย X เมื่อต้องการเช็ค ในแอพพลิเคชันนี้ผู้เขียนจะสร้างเช็คบ็อกซ์ขึ้นมาใช้เองโดยเลือกใช้เครื่องหมายขีดถูกแทน วิธีการก็คือเราต้องสร้างภาพที่จะใช้แสดงเป็นเครื่องหมายขีดถูก และภาพที่จะแสดงในช่วงที่มีการคลิ๊กเม้าส์ด้วยทั้งหมด 3 ภาพ (ดูรูปที่ 4) ภาพทั้งสามภาพนี้จะถูกกำหนดให้กับ Property ค่า Picture ของคอนโทรลอะเรย์ชนิดอิมเมจชื่อ checked คอนโทรลอะเรย์ทั้งสามจะวางซ้อนกัน โพรซีเดอร์ที่ต้องเขียนเมื่อมีการคลิ๊กเม้าส์บนคอนโทรลอิมเมจก็เพียงแต่คอยกำหนดค่า Visible ของแต่ละภาพให้แสดงอย่างเหมาะสม คือเมื่อ ConfirmFlag มีค่าเป็น 0 เราจะต้องเซตให้มีค่าเป็น 1 แล้วแสดงอิมเมจรูป CKON.ICO นั่นคือต้องการเช็คค่าของเช็คบ็อกซ์ แต่ถ้า ConfirmFlag เป็น 1 อยู่แล้วก็จะแสดงอิมเมจรูป CKOFF.ICO แทน นอกจากนี้ในช่วงที่คลิ๊กเม้าส์ เราจะต้องแสดงอิมเมจรูป CKPRS.ICO เพื่อแสดงการคลิ๊กเม้าส์ ในเหตุการณ์ checked__MouseDown( ) และเมื่อเลิกคลิ๊กเม้าส์ก็จะเรียกโพรซีเดอร์ checked__Click( ) มาทำงานเช่นกัน (ดูรูปที่ 9)



```
Sub checked_Click (Index As Integer)
  If ConfirmFlag = 0 Then
    ConfirmFlag = 1
    checked(0).Visible = False
    checked(1).Visible = True
    checked(2).Visible = False
  Else
    ConfirmFlag = 0
    checked(0).Visible = True
    checked(1).Visible = False
    checked(2).Visible = False
  End If
End Sub

Sub checked_MouseDown (Index As Integer, Button As Integer, <--
Shift As Integer, X As Single, Y As Single)
  checked(0).Visible = False
  checked(1).Visible = False
  checked(2).Visible = True
End Sub

Sub checked_MouseUp (Index As Integer, Button As Integer, <--
Shift As Integer, X As Single, Y As Single)
  checked_Click (Index)
End Sub
```

รูปที่ 9

```
Declare Sub GetWindowRect Lib "User" (ByVal hWnd As Integer, <--
lpRect As RECT)
Declare Sub SetBkColor Lib "GDI" (ByVal hDC As Integer, <--
ByVal crColor As Long)
Declare Sub Rectangle Lib "GDI" (ByVal hDC As Integer, <--
ByVal X1 As Integer, ByVal Y1 As Integer, <--
ByVal X2 As Integer, ByVal Y2 As Integer)
Declare Sub DeleteObject Lib "GDI" (ByVal hObject As Integer)

Declare Function GetDC Lib "User" (ByVal hWnd As Integer) <--
As Integer
Declare Function ReleaseDC Lib "User" (ByVal hWnd As Integer, <--
ByVal hDC As Integer) As Integer
Declare Function CreateSolidBrush Lib "GDI" (ByVal crColor As <--
Long) As Integer
Declare Function SelectObject Lib "GDI" (ByVal hWnd As <--
Integer, ByVal hObject As Integer) As Integer
```

รูปที่ 10

โพรซีเดอร์ที่ใช้แสดงการระเบิดของฟอร์ม

การเขียนเอฟเฟคท์ระเบิดให้ฟอร์มในช่วงที่เปิดฟอร์มนั้นเป็นเทคนิคที่ผู้อ่านสามารถนำไปใช้กับแอพพลิเคชันอื่นได้ แต่การเขียนโปรแกรมจะต้องใช้วินโดวส์ API หลาย ๆ ฟังก์ชัน ดังนั้นจะต้องประกาศฟังก์ชันแต่ละอันดังแสดงในรูปที่ 10

สำหรับซับรูทีน GetWindowRect จะใช้ไทพ์ RECT เพื่อกำหนดตัวแปรแบบผ่านแอดเดรส lpRect การกำหนดไทพ์ RECT ต้องทำไว้ในโกลบอลโมดูล DRAGDEL.BAS

```
Sub ExplodeForm (F As Form)
Const STEPS = 75
Dim FRect As RECT
Dim FullWidth As Integer, FullHeight As Integer
Dim I As Integer
Dim X As Integer, Y As Integer, Cx As Integer, Cy As Integer
Dim hDCScreen As Integer, hBrush As Integer, hOldBrush

  GetWindowRect F.hWnd, FRect
  FullWidth = FRect.Right - FRect.Left
  FullHeight = FRect.Bottom - FRect.Top
  hDCScreen = GetDC(0)
  hBrush = CreateSolidBrush(F.BackColor)
  hOldBrush = SelectObject(hDCScreen, hBrush)
For I = 1 To STEPS
  Cx = FullWidth * (I / STEPS)
  Cy = FullHeight * (I / STEPS)
  X = FRect.Left + (FullWidth - Cx) / 2
  Y = FRect.Top + (FullHeight - Cy) / 2
  Rectangle hDCScreen, X, Y, X + Cx, Y + Cy
Next I
If ReleaseDC(0, hDCScreen) = 0 Then
  MsgBox "Unable to Release Device Context", 16,      <--
"Drag and Drop"
  End If
  DeleteObject (hBrush)
  F.Visible = True
End Sub
```

รูปที่ 11

```
Type RECT
        Left As Integer
        Top As Integer
        Right As Integer
        Bottom As Integer
End Type
```

การแสดงเอฟเฟคท์ใช้ซับรูทีน ExplodeForm โดยเริ่มจากหาขนาดความกว้างและสูงของฟอร์มก่อน โดยเรียก GetWindowRect( ) หลังจากนั้นจะต้องจองการใช้ดีไวซ์คอนเท็กซ์ก่อนวาดภาพ และสร้างออปเจ็กต์แปรงทาสีแบคกราวนด์ของฟอร์ม จากนั้นก็จะเริ่มวาดรูปสี่เหลี่ยมด้วย Rectangle( ) โดยเริ่มจากขนาดเล็กที่สุด แล้วเพิ่มขนาดสี่เหลี่ยมโดยคำนวณค่า Cx, Cy ใหม่ทุกครั้งเท่า ๆ กัน สี่เหลี่ยมที่จะวาดจะมีทั้งหมด 75 รูป เมื่อวาดเสร็จก็ต้องยกเลิกการใช้ดีไวซ์คอนเท็กซ์และออปเจ็กต์แปรงด้วย (ดูรูปที่ 11)

จบอีกหนึ่งตอน

เทคนิคแดร็กแอนด์ดร็อบเป็นเทคนิคที่นิยมใช้ในแอพพลิเคชันบนวินโดวส์ ผู้เขียนได้นำเทคนิคการเขียนโปรแกรมแบบง่าย ๆ มาแสดงในบทความนี้ หวังว่าผู้อ่านคงใช้ประโยชน์จากเทคนิคการเขียนโปรแกรมแบบต่าง ๆ ไปใช้ได้อย่างสนุกนะครับ

หมายเหตุ

ในส่วนโค้ดของโปรแกรมในบทความนี้ที่มีเครื่องหมาย <-- แสดงท้ายบรรทัดจะหมายความว่า ผู้อ่านจะต้องพิมพ์โค้ดในบรรทัดถัดไปไว้ในบรรทัดเดียวกันทั้งหมด ขณะเขียนโค้ดในวิชวลเบสิก

หนังสือดี ซีเอ็ด

การพัฒนาระบบฐานข้อมูลด้วยภาษาซี

■ โดย ออล สตีเวนส์
เรียบเรียงโดย อนุชิต สุเมธิวิทย์/วีร์ ตั้งมั่นภักดีพงษ์
ราคาเล่มละ 150 บาท

หนังสือเล่มนี้ประกอบด้วยพื้นฐานและการออกแบบข้อมูลเบื้องต้น วิธีการสร้างระบบจัดการฐานข้อมูลอย่างง่ายด้วยภาษาซี พร้อมโปรแกรมที่สามารถนำไปใช้งานได้จริง ผู้อ่านจึงควรมีความสามารถในการเขียนโปรแกรมด้วยภาษาซีอยู่บ้างแล้ว นอกจากนี้จุดประสงค์ของหนังสือต้องการให้ผู้อ่านสามารถสร้าง tool เพื่อลดเวลาในการโปรแกรมลงได้อีกด้วย หนังสือเล่มนี้จึงเหมาะสำหรับผู้ที่สนใจเรื่องฐานข้อมูล และผู้ที่รักการเขียนโปรแกรมทั่วไป

หาซื้อได้ตามร้านหนังสือทั่วไป
หรือสั่งซื้อโดยตรงได้ที่

บริษัท ซีเอ็ดยูเคชั่น จำกัด
800/43-45 ซอยตระกูลสุข ถนนอโศก-ดินแดง ห้วยขวาง กรุงเทพฯ 10400 โทร. 248-6280-9 (เวลาทำการ จันทร์-ศุกร์ 8.10-17.30 น.) โดยส่งธนาณัติสั่งจ่าย ปท.ดินแดง ในนามบริษัทฯ หรือเช็คส่วนตัวของธนาคารในเขตกรุงเทพฯ พร้อมค่าจัดส่ง
ค่าจัดส่ง สั่งซื้อไม่เกิน 100 บาท ค่าจัดส่ง 8 บาท สั่งซื้อเกิน 100 บาท ค่าจัดส่ง 8%
พิเศษ สำหรับสมาชิกวารสารของซีเอ็ด สั่งซื้อโดยตรงที่บริษัทฯ ได้ในราคาลด 15% เพิ่มค่าลงทะเบียน 4 บาท ค่าจัดส่งฟรี โดยระบุหมายเลขสมาชิกด้วย